VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

PROJECT REPORT

Object Oriented Analysis and Design (ITE1007)

MEMBERS

SIDDHARTH DAS (18BIT0379)

RAHUL RAJ MISHRA (18BIT0380)

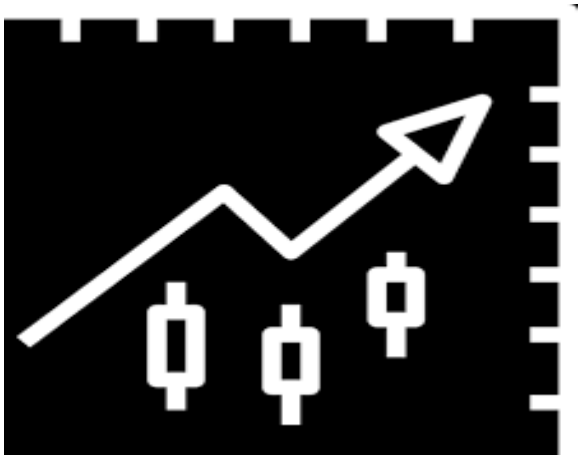
SHRUTI VARSHA VENKATRAMAN (18BIT0405)

FACULTY

PROF. BHAVANI. S

PROJECT TITLE

An Online Stock Brokerage System Design



Online Stock Broker

CONTENTS PAGES

Acknowledgement

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude to **Prof. Bhavani.S**, our OOAD Professor of Vellore Institute of Technology for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

Abstract

An Online Stock Brokerage System facilitates its users the trade (i.e. buying and selling) of stocks online. It allows clients to keep track of and execute their transactions, and shows performance charts of the different stocks in their portfolios. It also provides security for their transactions and alerts them to pre-defined levels of changes in stocks, without the use of any middlemen.

The online stock brokerage system automates traditional stock trading using computers and the internet, making the transaction faster and cheaper. This system also gives speedier access to stock reports, current market trends, and real-time stock prices.

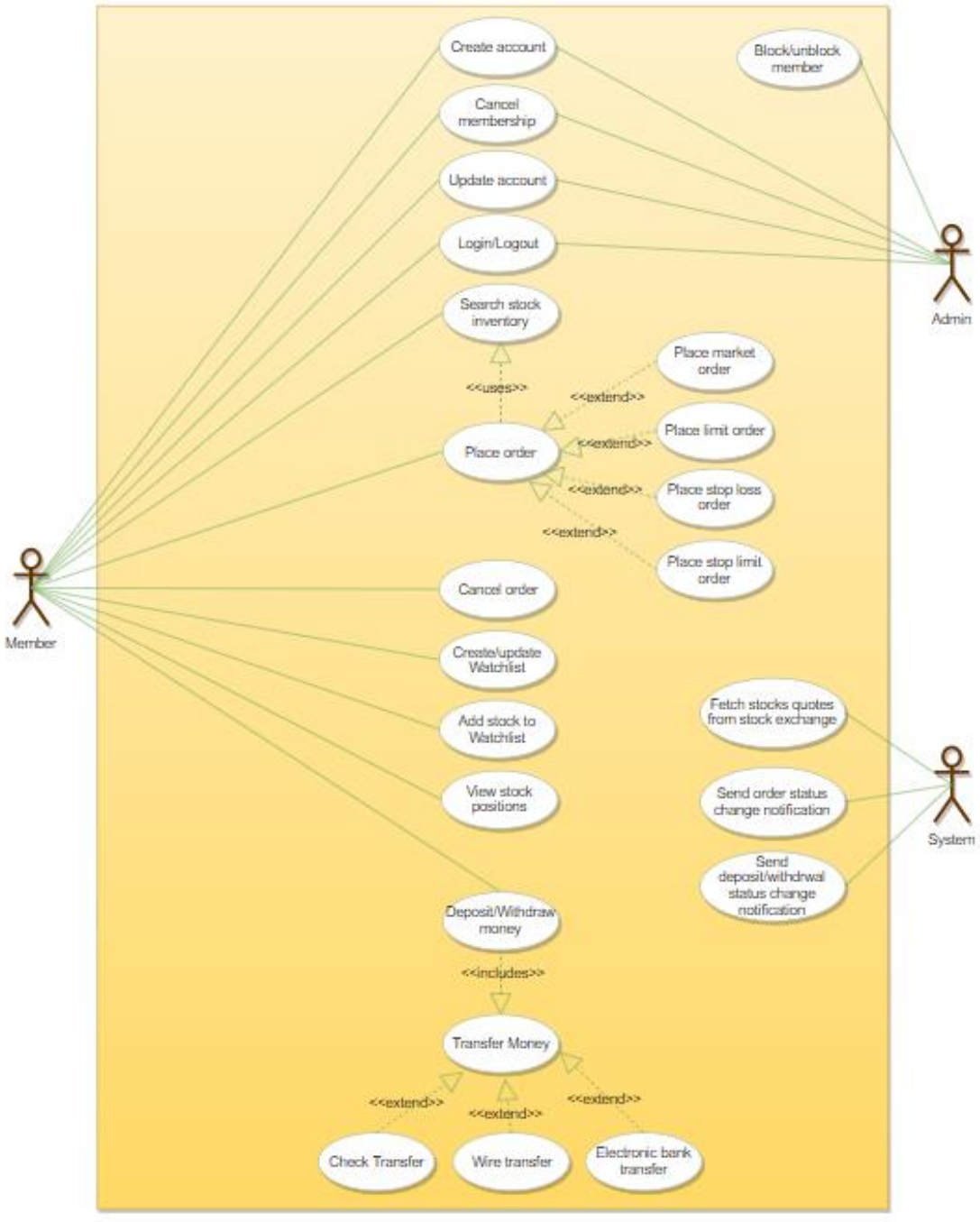
USE-CASE DIAGRAM

We have three main Actors in our system:

- **Admin:** Mainly responsible for administrative functions like blocking or unblocking members.
- **Member:** All members can search the stock inventory, as well as buy and sell stocks. Members can have multiple watchlists containing multiple stock quotes.
- **System:** Mainly responsible for sending notifications for stock orders and periodically fetching stock quotes from the stock exchange.

Here are the top use cases of the Stock Brokerage System:

- **Register new account/Cancel membership:** To add a new member or cancel the membership of an existing member.
- **Add/Remove/Edit watchlist:** To add, remove or modify a watchlist.
- **Search stock inventory:** To search for stocks by their symbols.
- **Place order:** To place a buy or sell order on the stock exchange.
- **Cancel order:** Cancel an already placed order.
- **Deposit/Withdraw money:** Members can deposit or withdraw money via check, wire or electronic bank transfer.



Use case diagram for Stock Brokerage System

CLASS DIAGRAM

Here are the main classes of our Online Stock Brokerage System:

Account: Consists of the member's name, address, e-mail, phone, total funds, funds that are available for trading, etc. We'll have two types of accounts in the system: one will be a general member, and the other will be an Admin. The Account class will also contain all the stocks the member is holding.

StockExchange: The stockbroker system will fetch all stocks and their current prices from the stock exchange. StockExchange will be a singleton class encapsulating all interactions with the stock exchange. This class will also be used to place stock trading orders on the stock exchange.

Stock: The basic building block of the system. Every stock will have a symbol, current trading price, etc.

StockInventory: This class will fetch and maintain the latest stock prices from the StockExchange. All system components will read the most recent stock prices from this class.

Watchlist: A watchlist will contain a list of stocks that the member wants to follow.

Order: Members can place stock trading orders whenever they would like to sell or buy stock positions. The system would support multiple types of orders:

Market Order: Market order will enable users to buy or sell stocks immediately at the current market price.

Limit Order: Limit orders will allow a user to set a price at which they want to buy or sell a stock.

Stop Loss Order: An order to buy or sell once the stock reaches a certain price.

Stop Limit Order: The stop-limit order will be executed at a specified price, or better, after a given stop price has been reached. Once the stop price is reached, the stop-limit order becomes a limit order to buy or sell at the limit price or better.

OrderPart: An order could be fulfilled in multiple parts. For example, a market order to buy 100 stocks could have one part containing 70 stocks at $10 and another part with 30 stocks at $10.05.

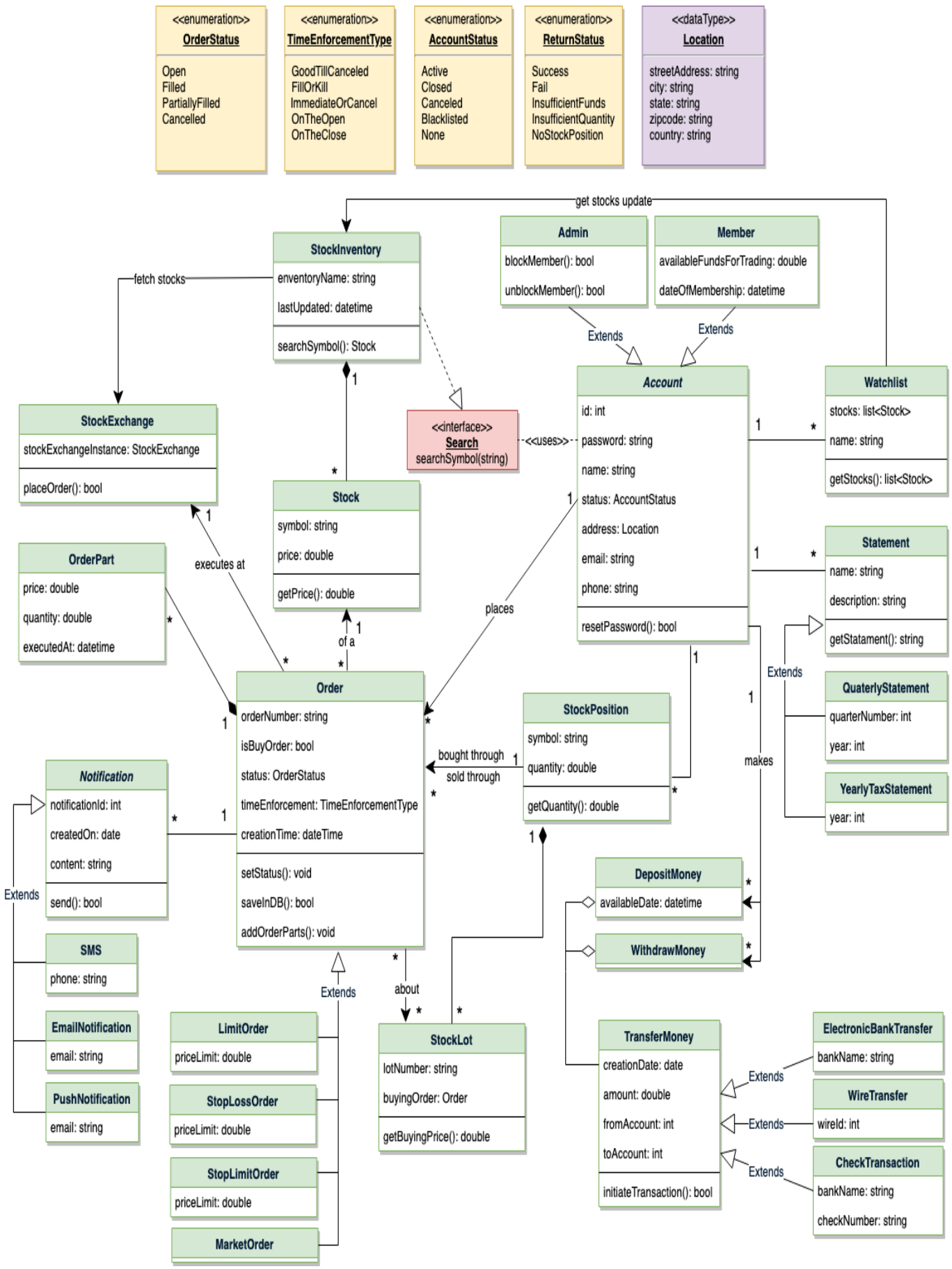
StockLot: Any member can buy multiple lots of the same stock at different times. This class will represent these individual lots. For example, the user could have purchased 100 shares of AAPL yesterday and 50 more stocks of AAPL today. While selling, users will be able to select which lot they want to sell first.

StockPosition: This class will contain all the stocks that the user holds.

Statement: All members will have reports for quarterly updates and yearly tax statements.

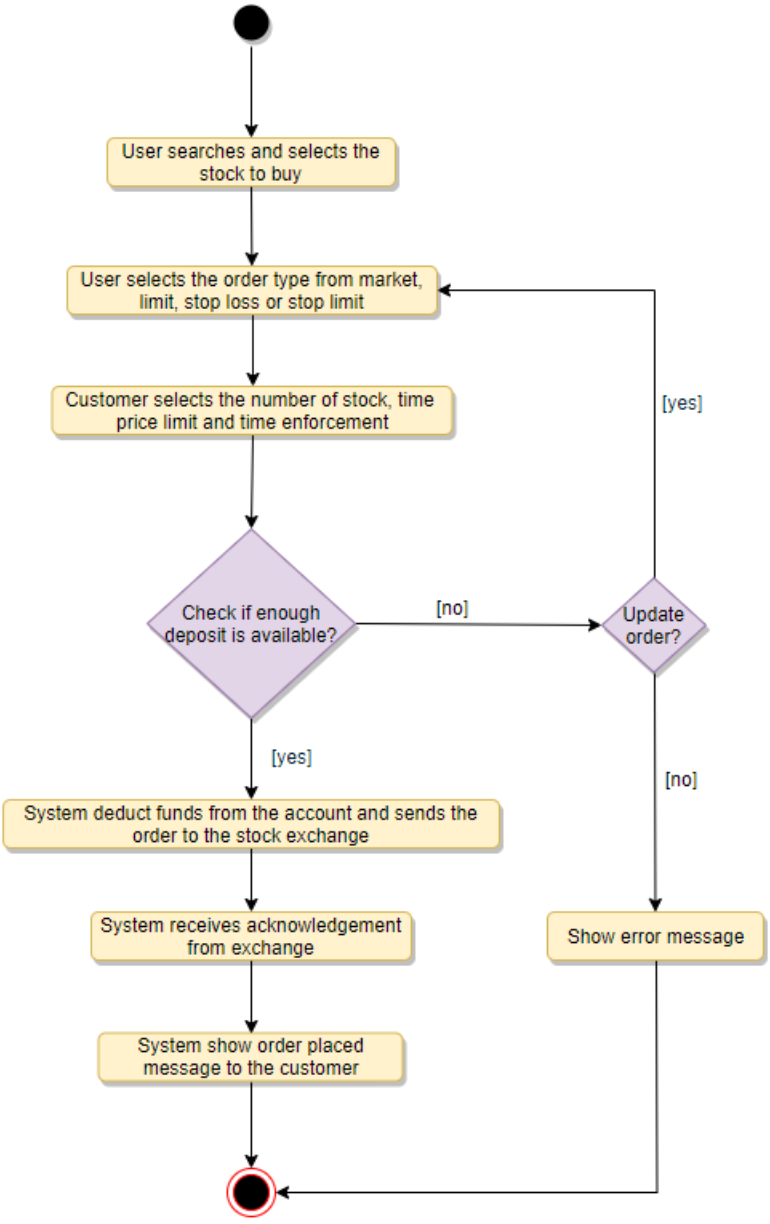
DepositMoney & WithdrawMoney: Members will be able to move money through check, wire or electronic bank transfers.

Notification: Will take care of sending notifications to members

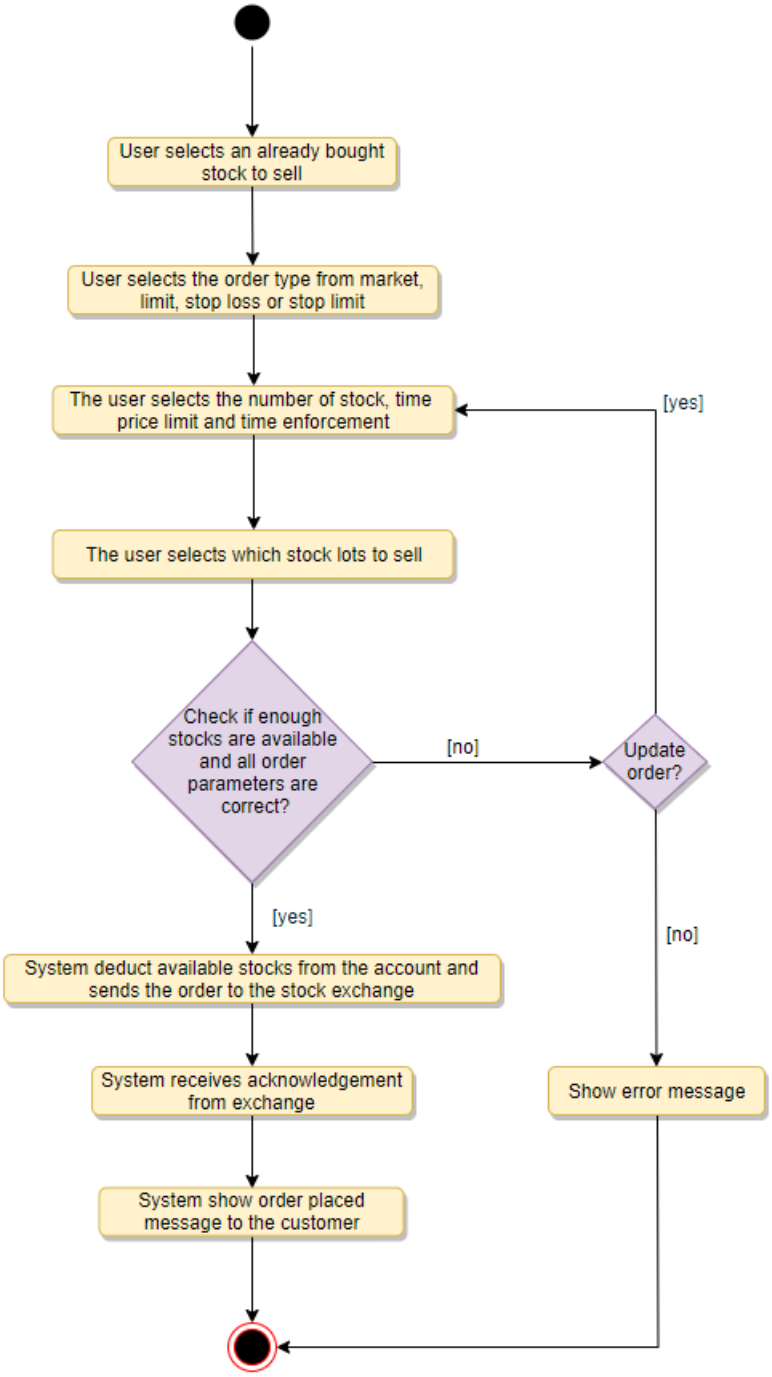

<<enumeration>>
OrderStatus
Open
Filled
PartiallyFilled
Cancelled
<<enumeration>>
TimeEnforcementType
GoodTillCanceled
FillOrKill
ImmediateOrCancel
OnTheOpen
OnTheClose
<<enumeration>>
AccountStatus
Active
Closed
Canceled
Blacklisted
None
<<enumeration>>
ReturnStatus
Success
Fail
InsufficientFunds
InsufficientQuantity
NoStockPosition
<<dataType>>
Location
streetAddress: string
city: string
state: string
zipcode: string
country: string
get stocks update
StockInventory
enventoryName: string
lastUpdated: datetime
searchSymbol(): Stock
fetch stocks
Admin
blockMember(): bool
unblockMember(): bool
Member
availableFundsForTrading: double
dateOfMembership: datetime
Extends
Extends
Account
id: int
password: string
name: string
status: AccountStatus
address: Location
email: string
phone: string
resetPassword(): bool
Watchlist
stocks: list<Stock>
name: string
getStocks(): list<Stock>
StockExchange
stockExchangeInstance: StockExchange
placeOrder(): bool
<<interface>>
Search
searchSymbol(string)
<<uses>>
Stock
symbol: string
price: double
getPrice(): double
executes at
OrderPart
price: double
quantity: double
executedAt: datetime
of a
places
Statement
name: string
description: string
getStatament(): string
Extends
QuaterlyStatement
quarterNumber: int
year: int
YearlyTaxStatement
year: int
Order
orderNumber: string
isBuyOrder: bool
status: OrderStatus
timeEnforcement: TimeEnforcementType
creationTime: dateTime
setStatus(): void
saveInDB(): bool
addOrderParts(): void
StockPosition
symbol: string
quantity: double
getQuantity(): double
bought through
sold through
makes
Notification
notificationId: int
createdOn: date
content: string
send(): bool
Extends
SMS
phone: string
EmailNotification
email: string
PushNotification
email: string
DepositMoney
availableDate: datetime
WithdrawMoney
Extends
about
LimitOrder
priceLimit: double
StopLossOrder
priceLimit: double
StopLimitOrder
priceLimit: double
MarketOrder
StockLot
lotNumber: string
buyingOrder: Order
getBuyingPrice(): double
TransferMoney
creationDate: date
amount: double
fromAccount: int
toAccount: int
initiateTransaction(): bool
Extends
Extends
Extends
ElectronicBankTransfer
bankName: string
WireTransfer
wireId: int
CheckTransaction
bankName: string
checkNumber: string

ACTIVITY DIAGRAM

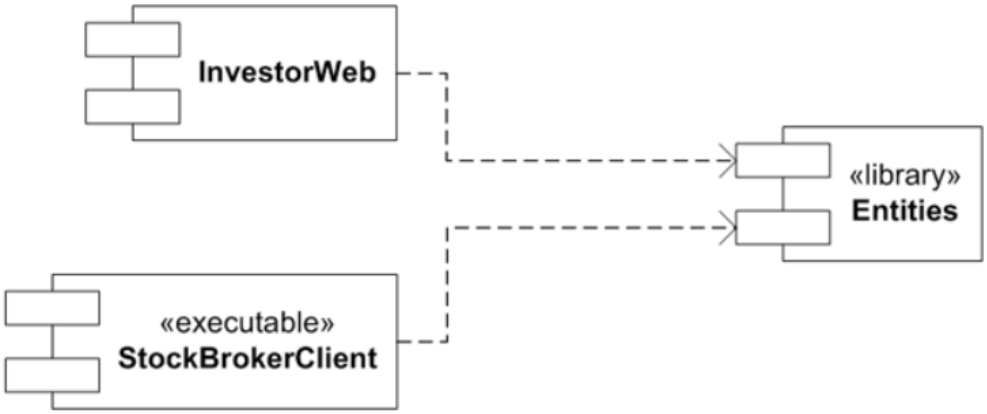
Place a buy order: Any system user can perform this activity. Here are the steps to place a buy order:



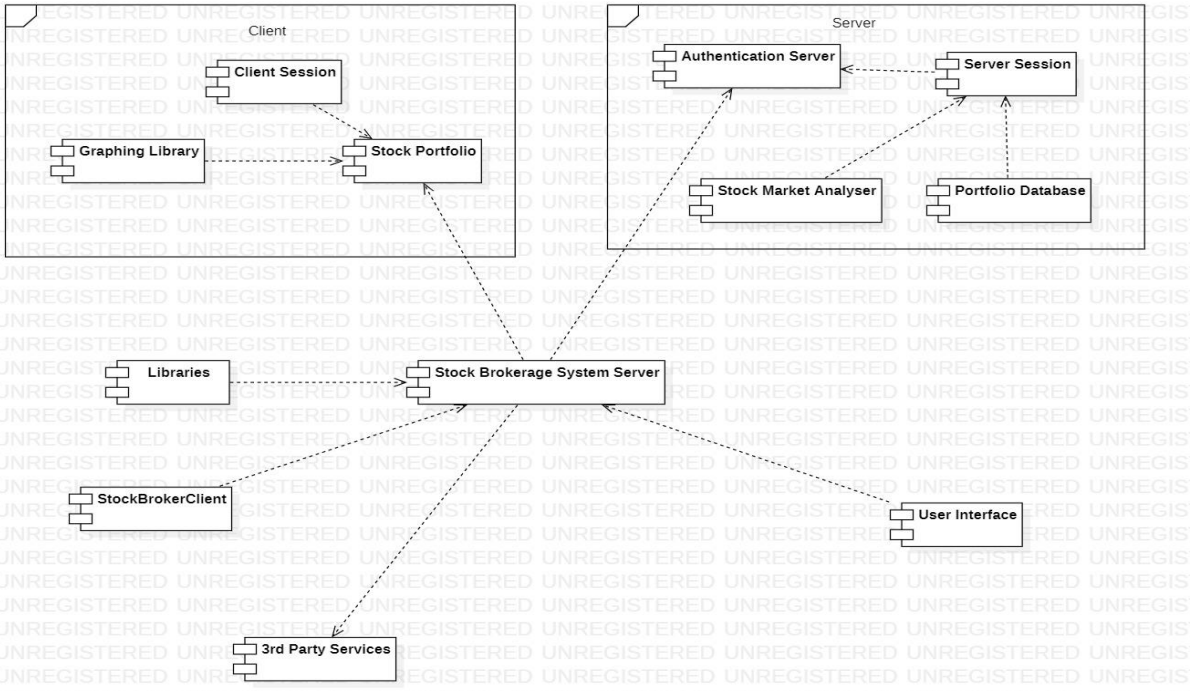
Place a sell order: Any system user can perform this activity. Here are the steps to place a buy order:



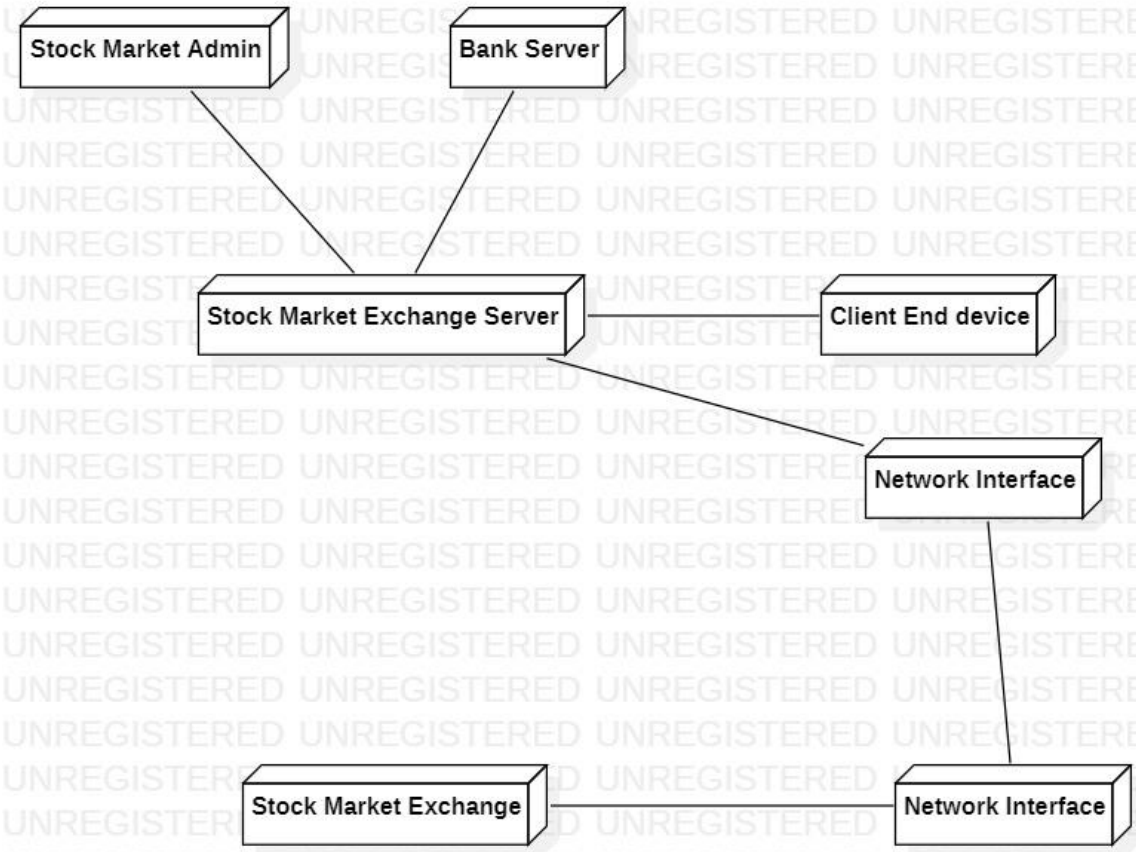
COMPONENT DIAGRAM:



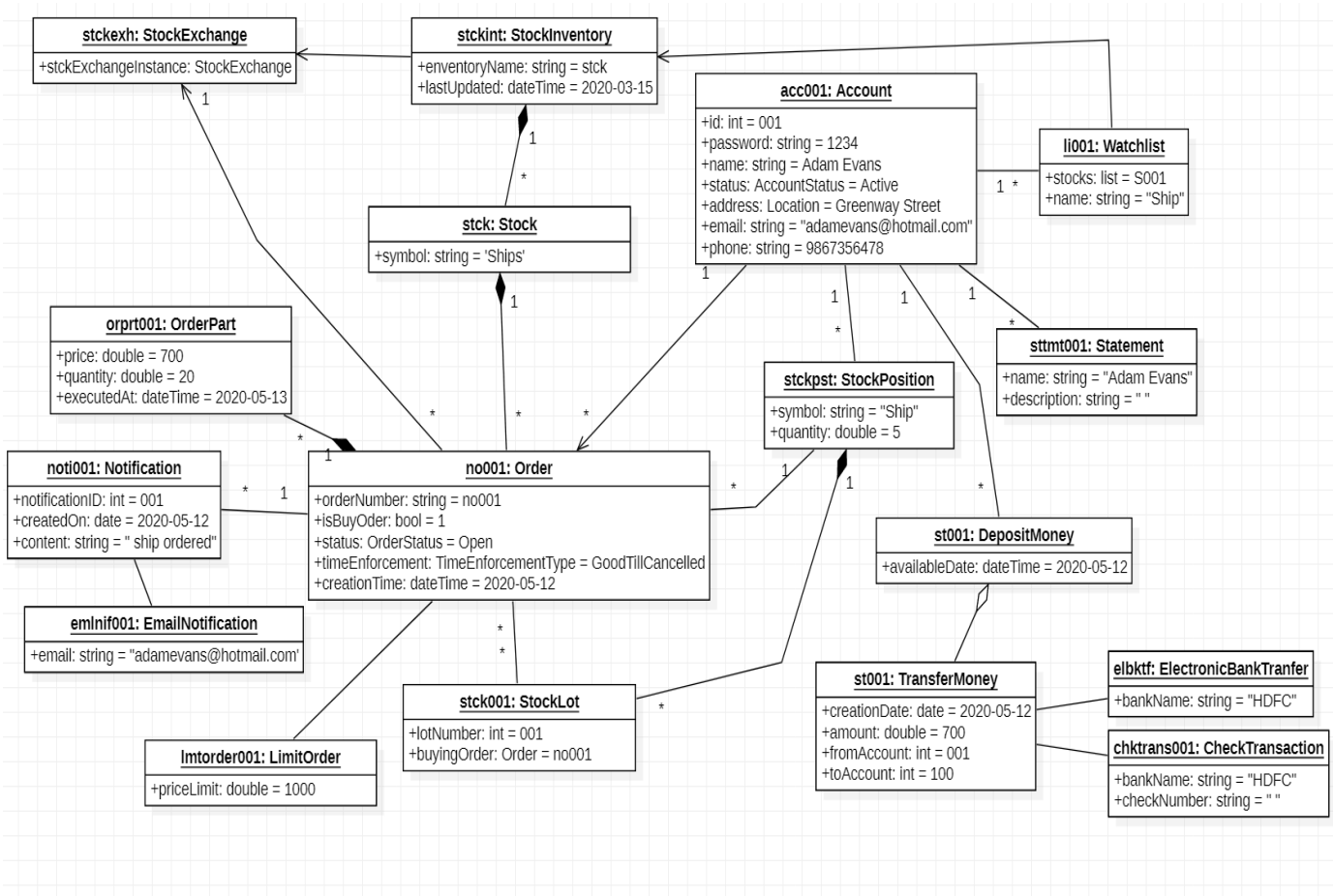
	InvestorWeb	StockBrokerClient	Entities
Entities: **AtBestOrder**			YES
Controllers: **ExecuteOrders**		YES	
Entities: **LimitOrder**			YES
Entities: **Order**			YES
Entities: **OrderBook**			YES
Forms: **OrderForm**	YES		
Controllers: **PlaceAtBestInstruction**	YES		
Controllers: **PlaceLimitOrderInstruction**	YES		
Controllers: **PlaceOrderInstruction**	YES		
Controllers: **PlaceStopOrderInstruction**	YES		
Controllers: **SelectStock**	YES		
Controllers: **SpecifyTargetPrice**	YES		
Forms: **StockList**	YES		
Entities: **StopOrder**			YES
Forms: **TargetPriceForm**	YES		



DEPLOYMENT DIAGRAM:

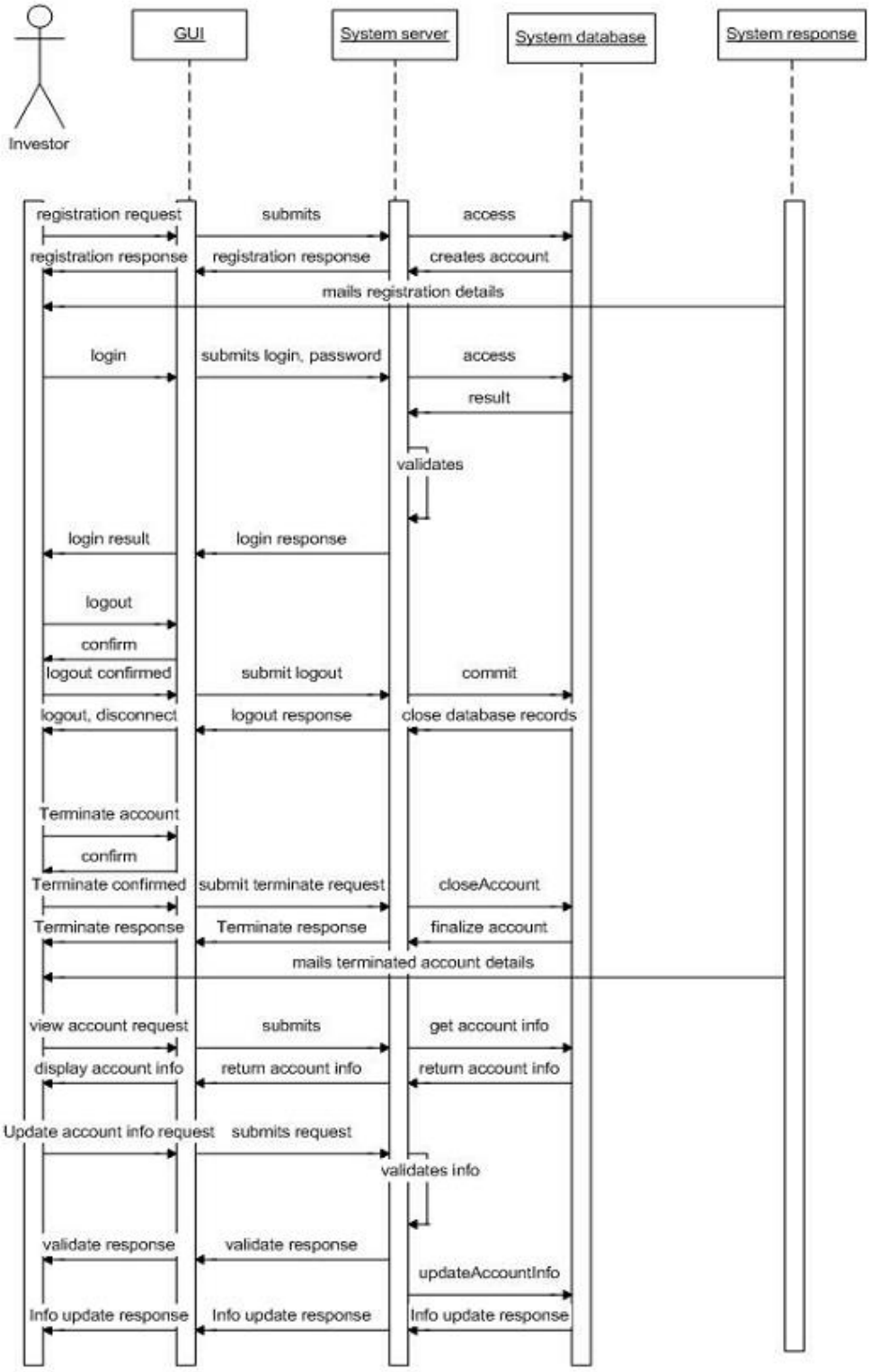


OBJECT DIAGRAM:

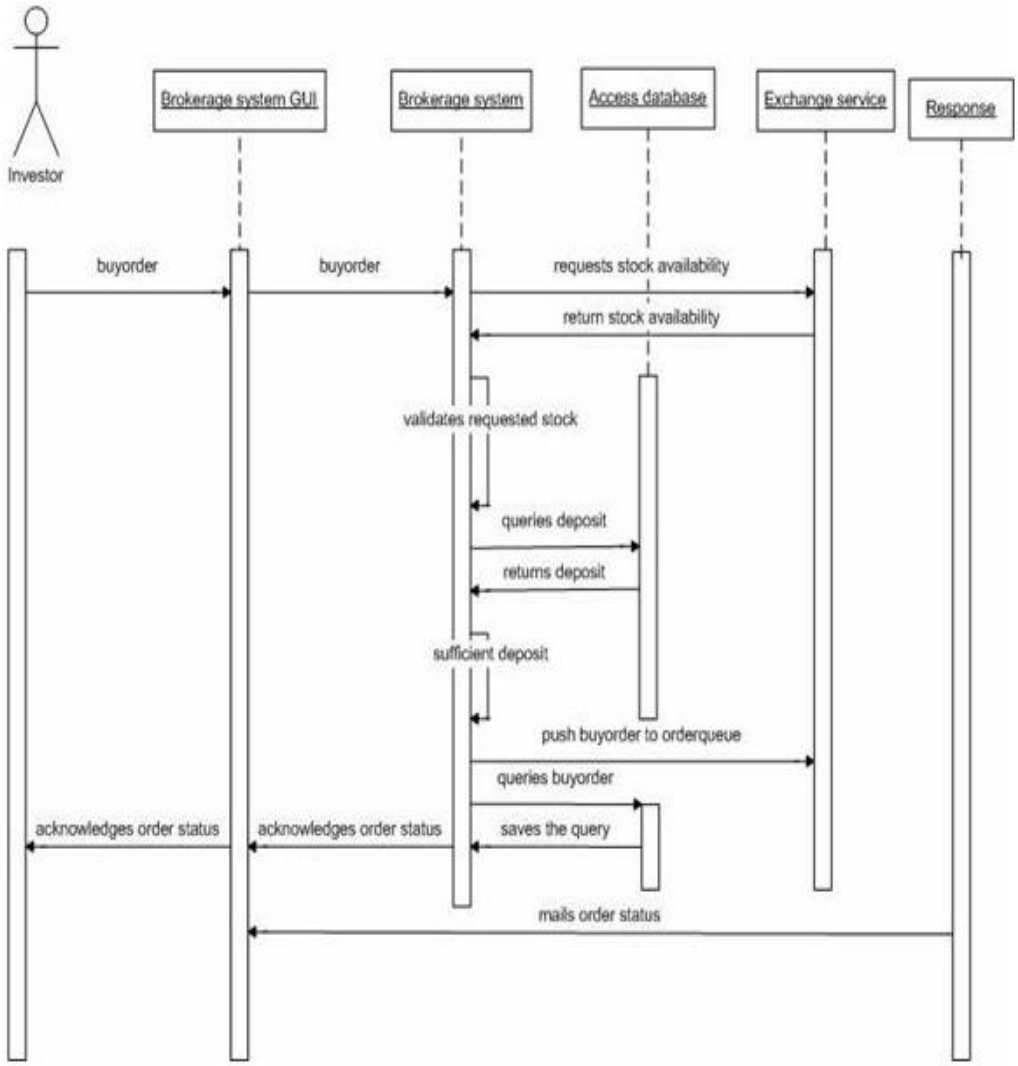


SEQUENCE DIAGRAM:

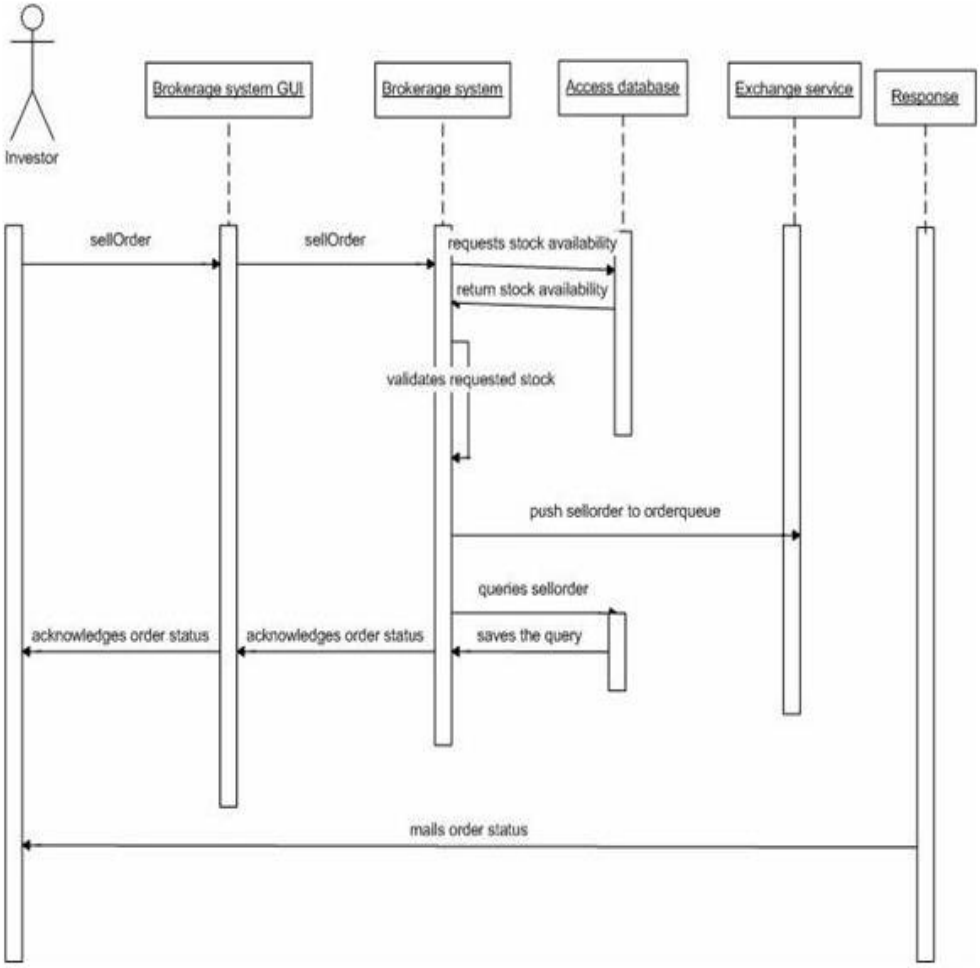
System behaviour for Client Activities



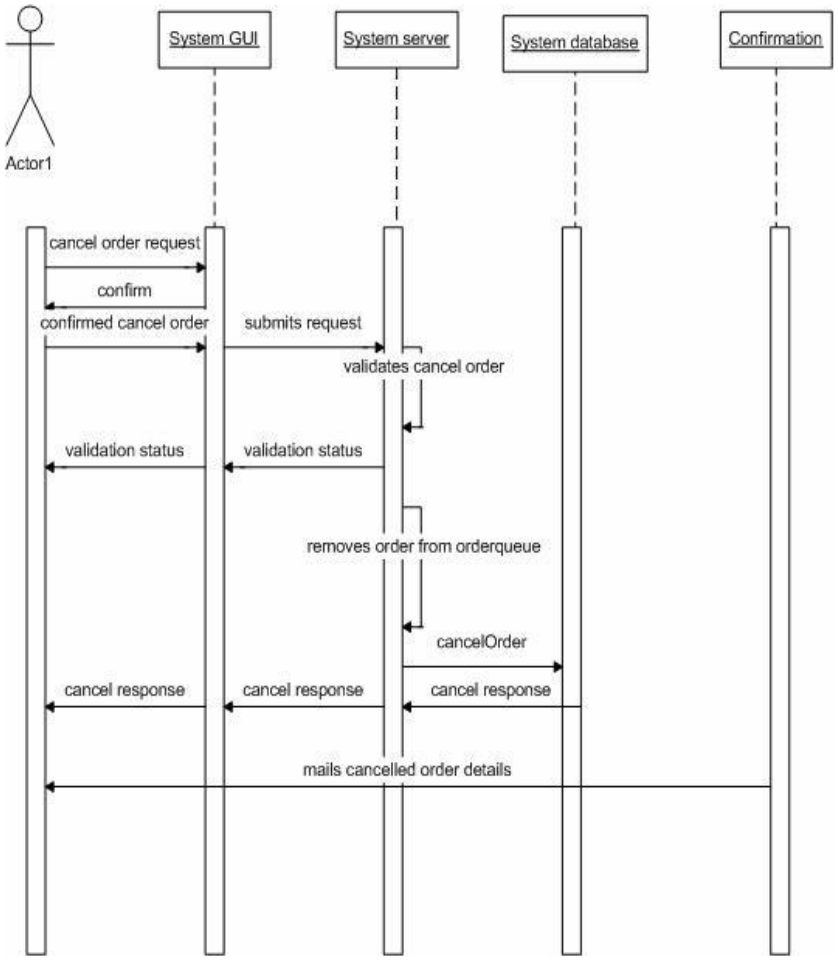
System Behaviour for Buying Stocks



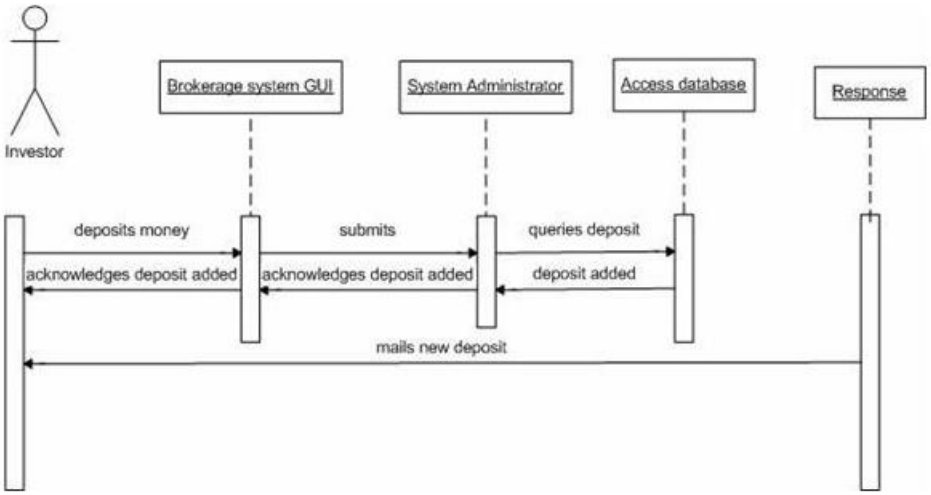
System Behaviour for Selling Stocks



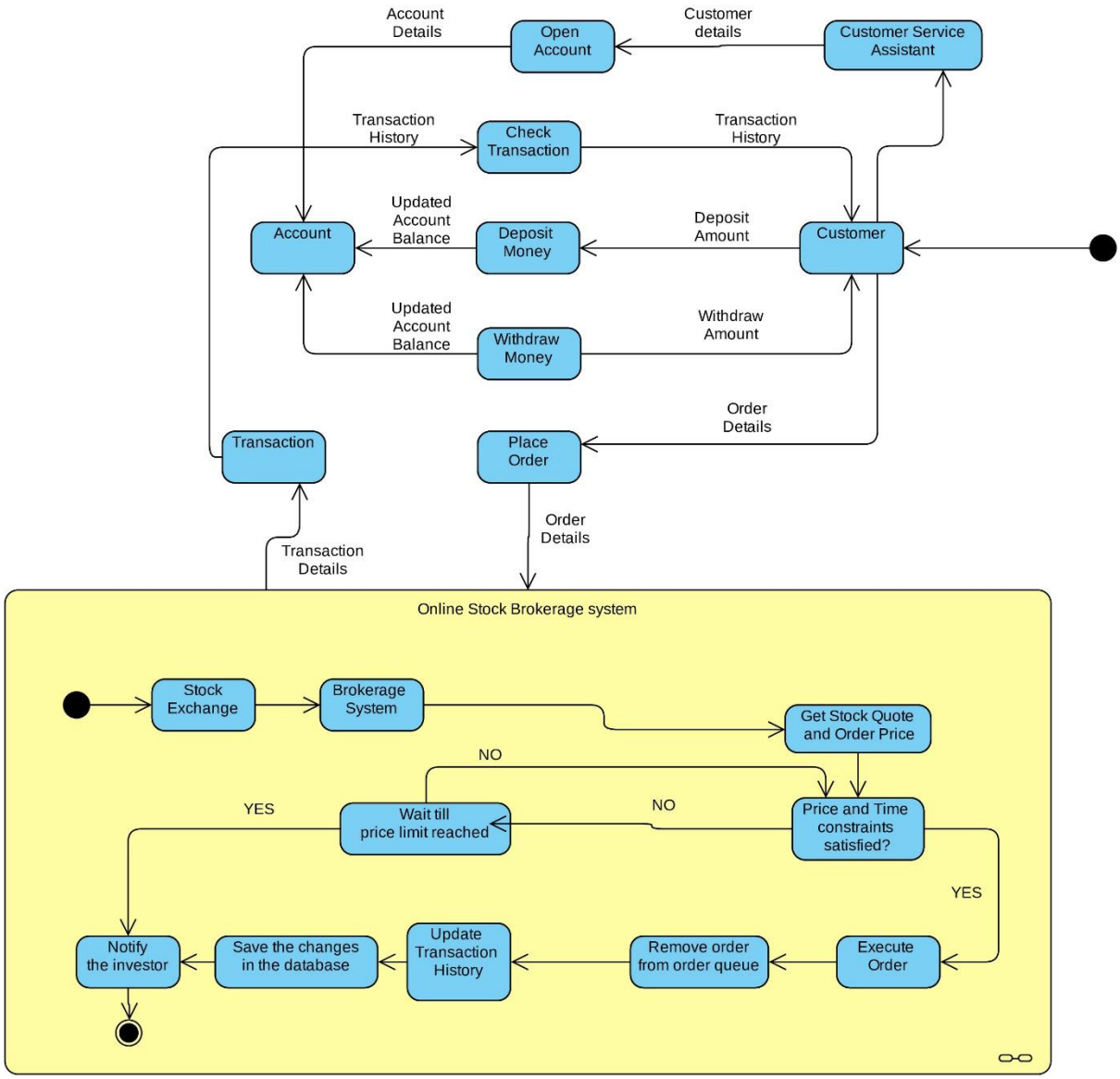
System Behaviour for Cancelling Order



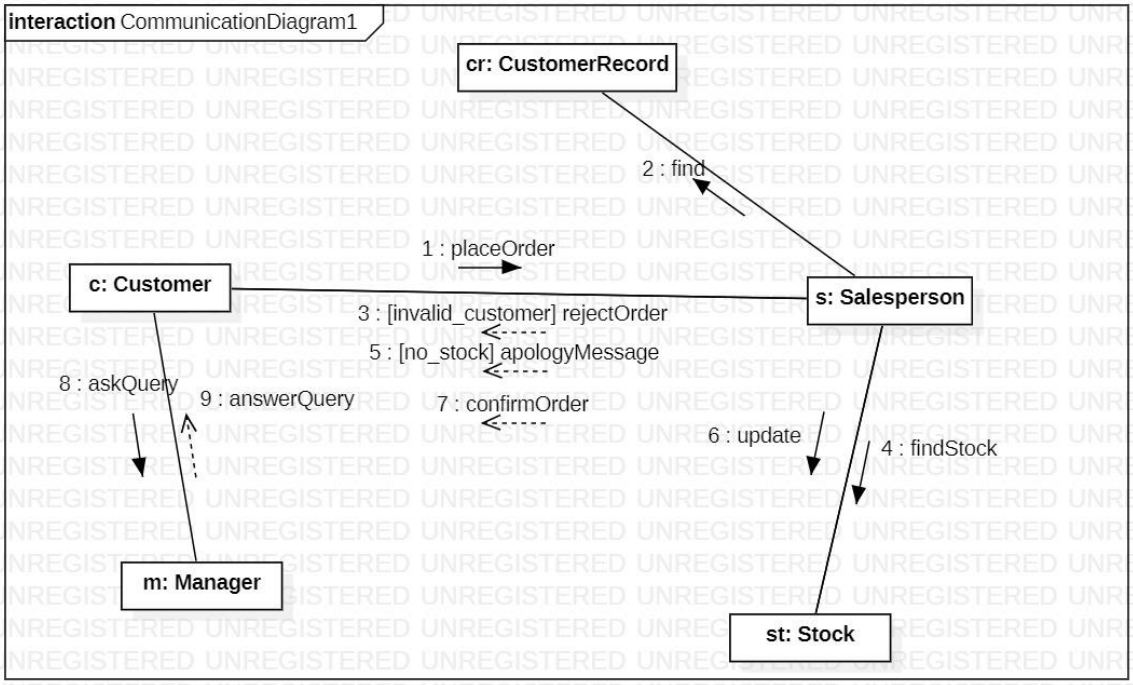
System Behaviour for Depositing Money



STATE-MACHINE DIAGRAM:



COLLABORATION DIAGRAM:



PRESENT SYSTEM VS PROPOSED SYSTEM

- **Description of Present System:**

Exiting system refers to the system that is being followed till now. Presently all the registrations are done manually. If a person wants to buy or sell stocks he should directly contact to the person who wish to sell or buy them. The main disadvantage is that there will be lot of difficulties for the citizens. So, all these procedures will be a time consuming one.

Limitations of Present System:

- Difficult for persons
- Time consuming

To avoid all these limitations and make the working more accurately the system needs to be computerized

- **Proposed System:**

Online Stock Brokerage Management System is aimed at developing a web-based system. In this the person can sales online and do many things. The details of the things are made available to them through the websites. Online Stock Brokerage Management System automates the traditional stock trading using computers and the internet, making the transaction faster and cheaper. This system also gives faster access to stock reports, current market trends and real time stock prices. The online stock brokerage system facilitates the users i.e., individual investors to trade (buy, sell, etc) the stocks online through the internet. It allows clients to keep track of, and execute their transactions, shows performance charts of different stocks in their portfolios, provides security for their transactions, and alerts them to pre-defined levels of changes in stocks, without the use of any middlemen.

Advantages of Proposed System:

- It improves the speed at which transactions are executed and further settled. This is due to the fact that there are no paper works to be documented, filed and then entered in an electronic format.
- This form of trading is quick and easy. You get the scope of educating yourself on the investment options that you have and then place orders for buying or selling. This enables you to make a substantial amount of income without having to speak with a broker and without leaving your house.
- It eliminates the need of getting in contact with a middleman. Previously, it was not possible to trade without getting hold of a middleman or a broker. But nowadays, it just takes a few clicks. It is this facility that makes online trades alluring especially for the people who do not have sufficient finances or connections of working with full service brokers. Online traders get the option of buying and selling easily without consulting a broker. This facility gives you the scope of trading virtually without any form of direct communication with a broker though there are always some indirect communications. This is because any form of trading is not possible without the use of a broker.
- Trading online offers good investor control. Online traders get the flexibility of trading whenever they want. This is not possible in traditional trading where an investor needs to delay the whole trade depending on the availability of a broker. Instantaneous transactions are the main features of online trades. Apart from this, investors are enabled to review the available options without having to depend on a broker for advising them on the best bets in return of their money.
- Traders who trade online can easily monitor their investments all through the day. The online brokerages provide advanced interfaces giving the investors the ability of seeing the performance of their money all through the day. The investors just need to log in using their computer or their phone and see their losses or gains in perfect real time. There are some online brokerages that also provide tools for all levels of traders. These brokerages post finance news and also provide research reports and analytic platforms.

SYSTEM REQUIREMENTS:

We will focus on the following set of requirements while designing the online stock brokerage system:

1. Any user of our system should be able to buy and sell stocks.
2. Any user can have multiple watchlists containing multiple stock quotes.
3. Users should be able to place stock trade orders of the following types:

   1) market,

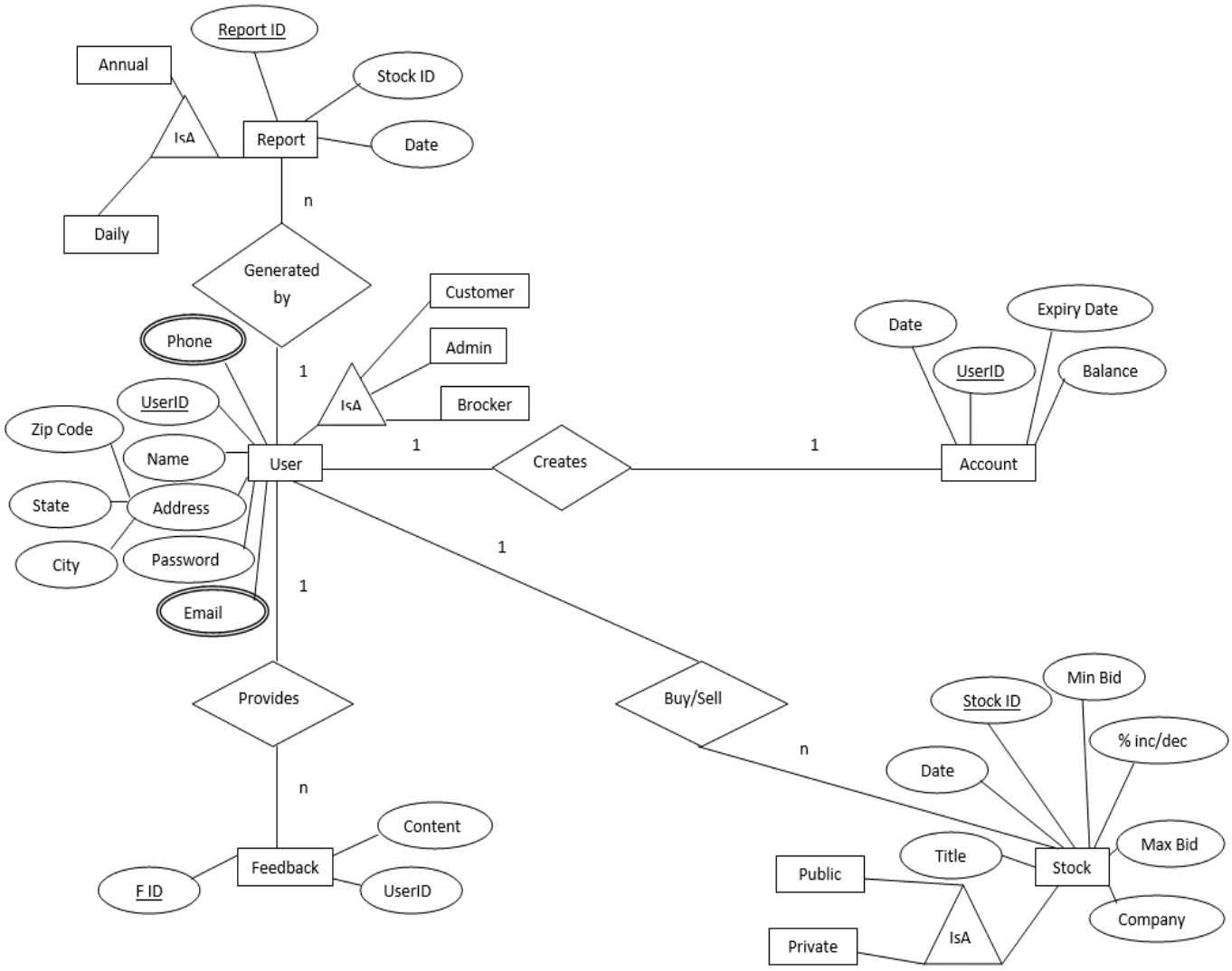
   2) limit,

   3) stop loss and,

   4) stop limit.
4. Users can have multiple ‘lots’ of a stock. This means that if a user has bought a stock multiple times, the system should be able to differentiate between different lots of the same stock.
5. The system should be able to generate reports for quarterly updates and yearly tax statements.
6. Users should be able to deposit and withdraw money either via check, wire or electronic bank transfer.
7. The system should be able to send notifications whenever trade orders are executed.

SYSTEM STRUCTURE:

- **Entity-Relationship Diagram:**



I. Structure of Tables:

- **Users:**

Field	Type	Size	Null	Key
USERID	NUMBER	38	NOT NULL	PRIMARY
NAME	VARCHAR2	50	NOT NULL	-
PHONE	VARCHAR2	20	NOT NULL	-
EMAIL	VARCHAR2	50	NOT NULL	-
PASSWORD	VARCHAR2	20	NOT NULL	-
ADDRESS	VARCHAR2	100	NOT NULL	-
USERTYPE	VARCHAR2	50	NOT NULL	-

- **Account:**

Field	Type	Size	Null	Key
USERID	NUMBER	38	NOT NULL	PRIMARY
ACC_NO	NUMBER	38	NOT NULL	PRIMARY
DDATE	DATE	-	NOT NULL	-
EXPIRY	DATE	-	NOT NULL	-
BALANCE	NUMBER	38	NOT NULL	-

- **Stock:**

Field	Type	Size	Null	Key
STOCKID	NUMBER	38	NOT NULL	PRIMARY
COMPANY	VARCHAR2	50	NOT NULL	-
TITLE	VARCHAR2	100	NOT NULL	-
DDATE	DATE	-	NOT NULL	-
MINBID	NUMBER	38	NOT NULL	-
MAXBID	NUMBER	38	NOT NULL	-
INC_DEC	NUMBER	38	NOT NULL	

- **Transaction:**

Field	Type	Size	Null	Key
TRANS_ID	NUMBER	38	NOT NULL	PRIMARY
USERID	NUMBER	38	NOT NULL	FOREIGN
DDATE	DATE	-	NOT NULL	-
STOCKID	NUMBER	38	NOT NULL	FOREIGN
QUANTITY	NUMBER	38	NOT NULL	-
PRICE_PER	NUMBER	38	NOT NULL	-
PRICE_TOTAL	NUMBER	38	NOT NULL	-

- **Feedback:**

Field	Type	Size	Null	Key
FID	NUMBER	38	NOT NULL	PRI
USERID	NUMBER	38	NOT NULL	FORIEGN
CONTENT	VARCHAR2	200	NOT NULL	-

DATABASE and CODES:

I. Creating and describing tables:

create table users(userid int not null, name varchar(50) not null,phone varchar(20) not null, email varchar(50) not null, password varchar(20) not null, address varchar(100) not null,usertype varchar(50) not null, primary key(userid));

```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(50)
 PHONE                                     NOT NULL VARCHAR2(20)
 EMAIL                                     NOT NULL VARCHAR2(50)
 PASSWORD                                  NOT NULL VARCHAR2(20)
 ADDRESS                                   NOT NULL VARCHAR2(100)
 USERTYPE                                  NOT NULL VARCHAR2(50)

SQL>
```

create table account (acc_no int not null, userid int not null,ddate date not null, expiry date not null, balance int not null, primary key(acc_no, userid));

```
SQL> desc account;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                    NOT NULL NUMBER(38)
 USERID                                    NOT NULL NUMBER(38)
 DDATE                                     NOT NULL DATE
 EXPIRY                                    NOT NULL DATE
 BALANCE                                   NOT NULL NUMBER(38)

SQL>
```

create table stock (stockid int not null, company varchar(50) not null, title varchar(100) not null,ddate date not null,minbid int not null,maxbid int not null,inc_dec int not null, primary key(stockid));

```
SQL> desc stock;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STOCKID                                   NOT NULL NUMBER(38)
 COMPANY                                   NOT NULL VARCHAR2(50)
 TITLE                                     NOT NULL VARCHAR2(100)
 DDATE                                     NOT NULL DATE
 MINBID                                    NOT NULL NUMBER(38)
 MAXBID                                    NOT NULL NUMBER(38)
 INC_DEC                                   NOT NULL NUMBER(38)

SQL>
```

create table transaction(trans_id int not null, userid int not null,ddate date not null,stockid int not null,quantity int not null,price_per int not null,price_total int not null,buy_sell varchar(50) not null, primary key(trans_id),foreign key(userid) references users(userid),foreign key(stockid) references stock(stockid));

```
SQL> desc transaction;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TRANS_ID                                  NOT NULL NUMBER(38)
 USERID                                    NOT NULL NUMBER(38)
 DDATE                                     NOT NULL DATE
 STOCKID                                   NOT NULL NUMBER(38)
 QUANTITY                                  NOT NULL NUMBER(38)
 PRICE_PER                                 NOT NULL NUMBER(38)
 PRICE_TOTAL                               NOT NULL NUMBER(38)
 BUY_SELL                                  NOT NULL VARCHAR2(50)

SQL>
```

create table feedback (fid int not null, userid int not null, content varchar(200) not null, primary(fid), foreign key(userid) references users(userid));

```
SQL> desc feedback;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FID                                       NOT NULL NUMBER(38)
 USERID                                    NOT NULL NUMBER(38)
 CONTENT                                   NOT NULL VARCHAR2(200)

SQL>
```

II. Creating sequences for auto incremental fields

```
create sequence user_seq start with 1 increment by 1 nocache nocycle;

create sequence useacc_seq start with 1 increment by 1 nocache nocycle;

create sequence acc_seq start with 1000 increment by 1 nocache nocycle;

create sequence stock_seq start with 1 increment by 1 nocache nocycle;

create sequence trans_seq start with 1 increment by 1 nocache nocycle;

create sequence feed_seq start with 1 increment by 1 nocache nocycle;
```

```
SQL> select * from user_sequences where sequence_name IN ('USER_SEQ','USEACC_SEQ','ACC_SEQ','STOCK_S
EQ','TRANS_SEQ','FEED_SEQ','REPORT_SEQ');

SEQUENCE_NAME                    MIN_VALUE  MAX_VALUE INCREMENT_BY C O CACHE_SIZE LAST_NUMBER
------------------------------ ---------- ---------- ------------ - - ---------- -----------
ACC_SEQ                                 1 1.0000E+27            1 N N          0        1000
FEED_SEQ                                1 1.0000E+27            1 N N          0           1
REPORT_SEQ                              1 1.0000E+27            1 N N          0           1
STOCK_SEQ                               1 1.0000E+27            1 N N          0           1
TRANS_SEQ                               1 1.0000E+27            1 N N          0           1
USEACC_SEQ                              1 1.0000E+27            1 N N          0           1
USER_SEQ                                1 1.0000E+27            1 N N          0           1

7 rows selected.

SQL>
```

III. Creating Views

```
create view user_details as select users.userid as userid,acc_no,name,phone,email,password,user type,address,ddate,expiry,balance from users inner join account on users.userid=account.userid;
```

```
SQL> select * from user_details;

   USERID     ACC_NO NAME                                               PHONE              EMAIL
--------- ---------- -------------------------------------------------- ------------------ ------
PASSWORD            USERTYPE
------------------- --------------------------------------------------
ADDRESS                                                                                    DDATE     EXPIRY       BALANCE
-------------------------------------------------------------------------------------------------------------------
        1       1001 sonali                                             9810555373         s2sonali@gmail.com
sonali110           admin
802,swami dayanand apt,sector56,dwarka                                                     03-NOV-15 02-NOV-17       1000

        2       1002 rahul                                              8860975492         r4rahul@gmail.com
rahul00             customer
43,vrindavan apt,sector22,gurgaon                                                          05-NOV-15 04-NOV-16       5000
```

Working of the System

ADMIN PANEL

I. Registering into System

```
insert into users values(user_seq.nextval,'sonali','9810555373','sonali@sonalichawla.com',
'sonali110','802,swami dayanand apt,sector56,dwarka','admin');

insert into account values(acc_seq.nextval,useacc_seq.nextval,'03-NOV-15','02-NOV-
17',1000);
```

```
SQL> select * from users;

   USERID NAME                                               PHONE              EMAIL                                              PASSWORD
--------- -------------------------------------------------- ------------------ --------------------------------------------------
ADDRESS                                                                                                                 USERTYPE
--------------------------------------------------------------------------------------------------------------------------------
        1 sonali                                             9810555373         sonali@sonalichawla.com                            sonali110
802,swami dayanand apt,sector56,dwarka                                                                                  admin
```

```
SQL> select * from account;

    ACC_NO     USERID DDATE     EXPIRY       BALANCE
---------- ---------- --------- --------- ----------
      1001          1 03-NOV-15 02-NOV-17       1000
```

Note: Every admin will be registered for 2years initially. After registering, admin will be redirected to the login panel.

II. Login into system

```
select * from users join account on users.name='sonali' and users.password='sonali110' and
users.userid=account.userid;
```

```
SQL> select * from users join account on users.name='sonali' and users.password='sonali110' and user
s.userid=account.userid;

   USERID NAME                                               PHONE              EMAIL                                              PASSWORD
--------- -------------------------------------------------- ------------------ --------------------------------------------------
ADDRESS                                                                                                                 USERTYPE
--------------------------------------------------------------------------------------------------------------------------------
  BALANCE
----------
        1 sonali                                             9810555373         sonali@sonalichawla.com                            sonali110
802,swami dayanand apt,sector56,dwarka                                                                                  admin
     1000
```

Note: If login credentials match with the entry, the admin is redirected to welcome panel. If login credentials do not match with the entry, the admin is redirected to the failure page.

III. Enter New Stock

```
insert into stock values(stock_seq.nextval,'State Bank of India','SBIN','30-Nov-15',70,75,2);
```

```
SQL> select * from stock;

   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE        MINBID     MAXBID    INC_DEC
----------------------------------------------------------------------------------------------------------------------------------------
         1 State Bank of India
SBIN                                                                                                 30-NOV-15         70         75          2
```

IV. Add new Admin

```
insert into users values(user_seq.nextval,'bhawna','8860975406', 'bhawna1668@gmail.com', 'bhawna','dlf phase 2,gurgaon','admin');

insert into account values(acc_seq.nextval,useacc_seq.nextval,'01-Dec-15','30-Nov-17',1000);
```

```
SQL> select * from users;

    USERID NAME                                               PHONE              EMAIL                                                PASSWORD
---------- -------------------------------------------------- ------------------ ------------------
ADDRESS                                                                                              USERTYPE
----------------------------------------------------------------------------------------------------------------------------------------
         4 bhawna                                             8860975406         bhawna1668@gmail.com                                 bhawna
dlf phase 2,gurgaon                                                                                  admin

         1 sonali                                             9810555373         s2sonali@gmail.com                                   sonali110
802,swami dayanand apt,sector56,dwarka                                                               admin

         2 rahul                                              8860975492         r4rahul@gmail.com                                    rahul00
43,vrindavan apt,sector22,gurgaon                                                                    customer


SQL> select * from account;

    ACC_NO     USERID DDATE     EXPIRY       BALANCE
---------- ---------- --------- --------- ----------
      1006          4 01-DEC-15 30-NOV-17       1000
      1001          1 03-NOV-15 02-NOV-17       1000
      1002          2 05-NOV-15 04-NOV-16       5000
```

V. Modify User Details

```
update users set email='s2sonali@gmail.com' where userid=1;
```

```
SQL> update users set email='s2sonali@gmail.com' where userid=1;

1 row updated.

SQL> select * from users;

    USERID NAME                                               PHONE              EMAIL                                                PASSWORD
---------- -------------------------------------------------- ------------------ ------------------
ADDRESS                                                                                              USERTYPE
----------------------------------------------------------------------------------------------------------------------------------------
         1 sonali                                             9810555373         s2sonali@gmail.com                                   sonali110
802,swami dayanand apt,sector56,dwarka                                                               admin

         2 rahul                                              8860975492         r4rahul@gmail.com                                    rahul00
43,vrindavan apt,sector22,gurgaon                                                                    customer
```

VI. Delete User

```
delete from users where userid=3;

delete from account where userid=3;
```

```
SQL> delete from users where userid=3;

1 row deleted.

SQL> delete from account where userid=3;

1 row deleted.
```

VII. View All Users

```
select * from user_details;
```

```
SQL> select * from user_details;

    USERID     ACC_NO NAME                                               PHONE               EMAIL
---------- ---------- -------------------------------------------------- ------------------- ------
PASSWORD             USERTYPE
-------------------- --------------------------------------------------
ADDRESS                                                                                      DDATE     EXPIRY       BALANCE
-------------------------------------------------------------------------------------------------------------------------

         1       1001 sonali                                             9810555373          s2sonali@gmail.com
sonali110            admin
802,swami dayanand apt,sector56,dwarka                                                       03-NOV-15 02-NOV-17       1000

         2       1002 rahul                                              8860975492          r4rahul@gmail.com
rahul00              customer
43,vrindavan apt,sector22,gurgaon                                                            05-NOV-15 04-NOV-16       5000
```

VIII. View Selected User

```
select * from user_details where userid=2;
```

```
SQL> select * from user_details where userid=2;

    USERID     ACC_NO NAME                                               PHONE               EMAIL
---------- ---------- -------------------------------------------------- ------------------- ------
PASSWORD             USERTYPE
-------------------- --------------------------------------------------
ADDRESS                                                                                      DDATE     EXPIRY       BALANCE
-------------------------------------------------------------------------------------------------------------------------

         2       1002 rahul                                              8860975492          r4rahul@gmail.com
rahul00              customer
43,vrindavan apt,sector22,gurgaon                                                            05-NOV-15 04-NOV-16       5000
```

IX. Modify Stock Details

update stock set minbid=65,inc_dec=3 where stockid=1;

```
SQL> update stock set minbid=65,inc_dec=3 where stockid=1;

1 row updated.

SQL> select * from stock where stockid=1;

   STOCKID COMPANY                                            TITLE
---------- -------------------------------------------------- ----------
   INC_DEC
----------
         1 State Bank of India                                SBIN
         3
```

X. Delete Stock

delete from stock where stockid=2;

```
SQL> delete from stock where stockid=2;

1 row deleted.
```

XI. View Stocks

select * from stock;

```
SQL> select * from stock;

   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE        MINBID     MAXBID    INC_DEC
----------------------------------------------------------------------------------------------------------------------------------------------
         1 State Bank of India
SBIN                                                                                                 30-NOV-15        65         75          3
```

XII. Buy/Sell Stock

insert into transaction values(trans_seq.nextval,1,'01-Nov-2015',1,1,65,65,'sell');

insert into transaction values(trans_seq.nextval,1,'01-Nov-2015',1,2,66,132,'buy');

```
SQL> select * from transaction;

  TRANS_ID     USERID DDATE       STOCKID   QUANTITY  PRICE_PER PRICE_TOTAL BUY_SELL
---------- ---------- --------- ---------- ---------- ---------- ----------- ----------
         1          2 30-NOV-15          1          2         71         142 buy
         2          2 30-NOV-15          1          1         75          75 sell
```

XIII. View My Transactions

```
select * from transaction where userid=2 order by trans_id desc;
```

```
SQL> select * from transaction where userid=1 order by trans_id desc;

  TRANS_ID     USERID DDATE        STOCKID   QUANTITY  PRICE_PER PRICE_TOTAL BUY_SELL
---------- ---------- --------- ---------- ---------- ---------- ----------- ------------------------
         4          1 01-NOV-15          1          2         66         132 buy
         3          1 01-NOV-15          1          1         65          65 sell
```

XIV. View All Transactions

```
select * from transaction order by trans_id desc;
```

```
SQL> select * from transaction order by trans_id desc;

  TRANS_ID     USERID DDATE        STOCKID   QUANTITY  PRICE_PER PRICE_TOTAL BUY_SELL
---------- ---------- --------- ---------- ---------- ---------- ----------- ------------------------
         4          1 01-NOV-15          1          2         66         132 buy
         3          1 01-NOV-15          1          1         65          65 sell
         2          2 30-NOV-15          1          1         75          75 sell
         1          2 30-NOV-15          1          2         71         142 buy
```

XV. View Stock Report

```
select * from stock where title='SBIN' and ddate between '30-Nov-15' and '5-Dec-15' order by st ockid desc;
```

```
SQL> select * from stock where title='SBIN' and ddate between '30-Nov-15' and '5-Dec-15' order by st
ockid desc;

   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE        MINBID     MAXBID    INC_DEC
---------------------------------------------------------------------------------------------------------------------------------------------
         5 State Bank of India
SBIN                                                                                                 03-DEC-15         76         95         10

         4 State Bank of India
SBIN                                                                                                 02-DEC-15         66         85          5

         3 State Bank of India
SBIN                                                                                                 01-DEC-15         60         70         -2


   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE        MINBID     MAXBID    INC_DEC
---------------------------------------------------------------------------------------------------------------------------------------------
         1 State Bank of India
SBIN                                                                                                 30-NOV-15         65         75          3
```

XVI. Give Feedback

```
insert into feedback values(feed_seq.nextval,1,'Excellent');
```

```
SQL> select * from feedback;

       FID     USERID
---------- ----------
CONTENT
--------------------------------------------------------------------------------
         2          1
Excellent
```

XVII. View Feedback

```
select * from feedback order by fid desc;
```

```
SQL> select * from feedback;

       FID     USERID
---------- ----------
CONTENT
--------------------------------------------------------------------------------
         1          2
Good
```

XVIII. Delete Feedback

```
delete from feedback where fid=1;
```

```
SQL> delete from feedback where fid=1;

1 row deleted.
```

CUSTOMER PANEL

I. Registering into System

```
insert into users values(user_seq.nextval,'rahul','8860975492','r4rahul@gmail.com',
'rahul00','43,vrindavan apt,sector22,gurgaon','customer');

insert into account values(acc_seq.nextval,useacc_seq.nextval,'05-NOV-15','04-NOV-
16',5000);
```

```
SQL> select * from users;

   USERID NAME                                               PHONE            EMAIL                                    PASSWORD
--------- -------------------------------------------------- ---------------- ----------------
ADDRESS                                                                                                 USERTYPE
--------------------------------------------------------------------------------------------------------------
        1 sonali                                             9810555373       sonali@sonalichawla.com                  sonali110
802,swami dayanand apt,sector56,dwarka                                                                  admin

        2 rahul                                              8860975492       r4rahul@gmail.com                        rahul00
43,vrindavan apt,sector22,gurgaon                                                                       customer
```

```
SQL> select * from account;

    ACC_NO     USERID DDATE     EXPIRY       BALANCE
---------- ---------- --------- --------- ----------
      1001          1 03-NOV-15 02-NOV-17       1000
      1002          2 05-NOV-15 04-NOV-16       5000
```

Note: Every customer will be registered for 1years initially. After registering, customer will be redirected to the login panel.

II. Login into system

```
select * from users join account on users.name='rahul' and users.password='rahul00' and users.userid=account.userid;
```

```
SQL> select * from users join account on users.name='rahul' and users.password='rahul00' and users.u
serid=account.userid;

    USERID NAME                                               PHONE              EMAIL                                              PASSWORD
---------- -------------------------------------------------- ------------------ --------------------
ADDRESS                                                                                                 USERTYPE
--------------------------------------------------------------------------------------------------------------
   BALANCE
----------
         2 rahul                                              88680975492        r4rahul@gmail.com                                  rahul00
43,vrindavan apt,sector22,gurgaon                                                                       customer
      5000
```

Note: If login credentials match with the entry, the customer is redirected to welcome panel. If login credentials do not match with the entry, the customer is redirected to the failure page.

III. View Stocks

```
select * from stock;
```

```
SQL> select * from stock;

   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                   DDATE       MINBID     MAXBID    INC_DEC
--------------------------------------------------------------------------------------------------------------
         1 State Bank of India
SBIN                                                                                                    30-NOV-15       65         75          3
```

IV. Buy/Sell Stock

```
insert into transaction values(trans_seq.nextval,2, '30-Nov-15',1,2,71,142, 'buy');

insert into transaction values(trans_seq.nextval,2, '30-Nov-15',1,1,75,75, 'sell');
```

```
SQL> select * from transaction;

  TRANS_ID     USERID DDATE        STOCKID   QUANTITY  PRICE_PER PRICE_TOTAL BUY_SELL
---------- ---------- --------- ---------- ---------- ---------- ----------- ----------
         1          2 30-NOV-15          1          2         71         142 buy
         2          2 30-NOV-15          1          1         75          75 sell
```

V. View My Transactions

```
select * from transaction where userid=2 order by trans_id desc;
```

```
SQL> select * from transaction where userid=2 order by trans_id desc;

  TRANS_ID     USERID DDATE        STOCKID   QUANTITY  PRICE_PER PRICE_TOTAL BUY_SELL
---------- ---------- --------- ---------- ---------- ---------- ----------- ------------------------
         2          2 30-NOV-15          1          1         75          75 sell
         1          2 30-NOV-15          1          2         71         142 buy
```

VI. Give Feedback

```
insert into feedback values(feed_seq.nextval,2,'Good');
```

```
SQL> select * from feedback;

       FID     USERID
---------- ----------
CONTENT
--------------------------------------------------------------------------------------------------------------------------------
         1          2
Good
```

VII. View Stock Report

```
select * from stock where title='SBIN' and ddate between '30-Nov-15' and '5-Dec-15' order by stockid desc;
```

```
SQL> select * from stock where title='SBIN' and ddate between '30-Nov-15' and '5-Dec-15' order by st
ockid desc;

   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE       MINBID    MAXBID    INC_DEC
----------------------------------------------------------------------------------------------------
         5 State Bank of India
SBIN                                                                                                 03-DEC-15       76        95         10

         4 State Bank of India
SBIN                                                                                                 02-DEC-15       66        85          5

         3 State Bank of India
SBIN                                                                                                 01-DEC-15       60        70         -2


   STOCKID COMPANY
---------- --------------------------------------------------
TITLE                                                                                                DDATE       MINBID    MAXBID    INC_DEC
----------------------------------------------------------------------------------------------------
         1 State Bank of India
SBIN                                                                                                 30-NOV-15       65        75          3
```

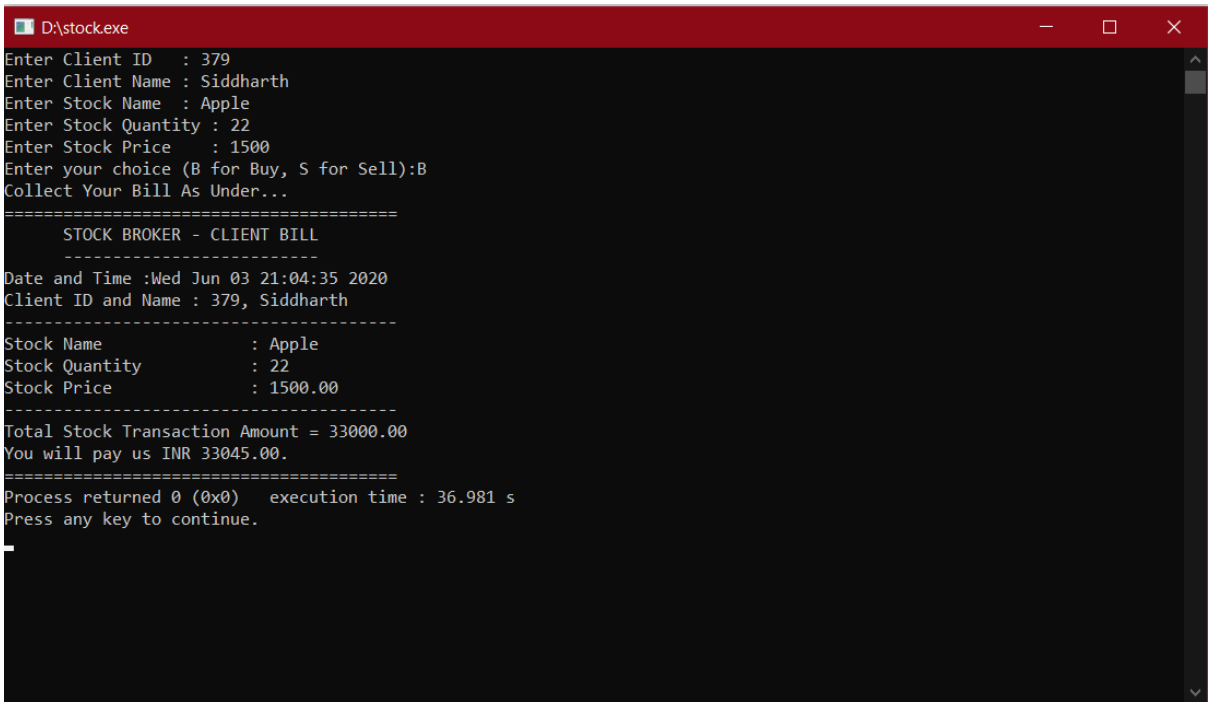
CLIENT BILLING - C PROGRAM

```
#include<stdio.h>
#include<time.h>
int main()
{
 int stock_quantity, client_id, brokerage = 45;
 float stock_price, total;
 char choice, stock_name[20], client_name[20];
 time_t t;
 time(&t);

 printf("Enter Client ID   : ");
 scanf("%d",&client_id);

 printf("Enter Client Name : ");
 scanf("%s", client_name);

 printf("Enter Stock Name  : ");
 scanf("%s", stock_name);

 printf("Enter Stock Quantity : ");
 scanf("%d", &stock_quantity);

 printf("Enter Stock Price    : ");
 scanf("%f", &stock_price);
```

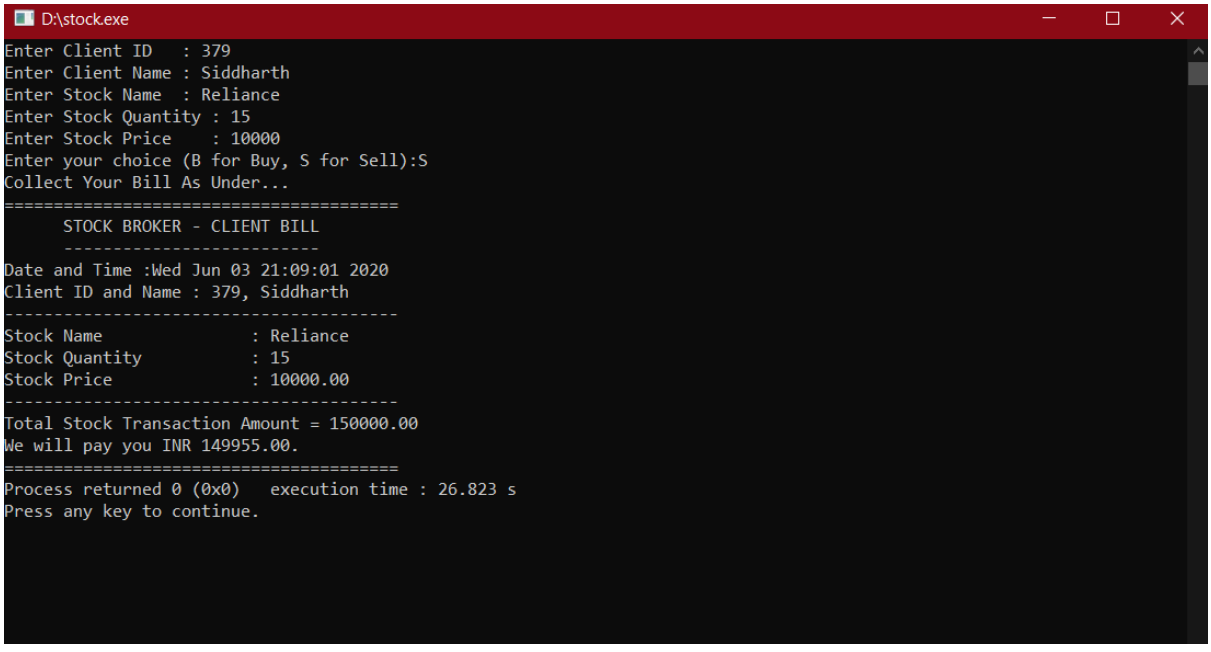
```
BACK:
 printf("Enter your choice (B for Buy, S for Sell):");
 scanf(" %c", &choice);
 if(choice!='B' && choice!='S')
 {
  printf("\n"); goto BACK;
 }
 total = stock_quantity * stock_price;
 printf("Collect Your Bill As Under...\n");
 printf("========================================\n");
 printf("      STOCK BROKER - CLIENT BILL\n");
 printf("      -------------------------\n");
 printf("Date and Time :%s",ctime(&t));
 printf("Client ID and Name : %d, %s\n",client_id, client_name);
 printf("----------------------------------------\n");
 printf("Stock Name\t\t : %s \n", stock_name);
 printf("Stock Quantity\t\t : %d\n", stock_quantity);
 printf("Stock Price\t\t : %0.2f\n", stock_price);
 printf("----------------------------------------\n");
 printf("Total Stock Transaction Amount = %0.2f\n", total);
 if(choice=='B')
  printf("You will pay us INR %0.2f.\n", total+brokerage);
 if(choice=='S')
  printf("We will pay you INR %0.2f.\n", total-brokerage);
 printf("========================================");

 return 0;
}
```

OUTPUT:

- **Buying stocks:**



- **Selling Stocks:**



CONCLUSION:

This venture gives us the opportunity to create an Online Stock Brokerage Management System planned for building up an electronic framework. This incorporates documentation of the proposed framework by utilizing prophet innovation as back-end. In this the individual can buy and sell stocks on the web and do numerous things. The subtleties of the things are made accessible to them through the sites. Online Stock Brokerage Management System computerizes the conventional stock exchanging utilizing PCs and the web, making the exchange quicker and less expensive. This framework additionally gives quicker access to stock reports, current market patterns and constant stock costs. The online stock financier framework encourages the clients i.e., singular speculators to exchange (purchase, sell, and so on) the stocks online through the web. It permits customers to monitor, and execute their exchanges, shows execution graphs of various stocks in their portfolios, gives security to their exchanges, and alarms them to pre-characterized levels of changes in stocks, without the utilization of any brokers.

Online stock brokerage system helps people to trade stocks through internet, without using middlemen, thus making the transactions faster than before. But the system has its own limitations. In this project, I have successfully applied the systems engineering methodology to design the brokerage system.

Applying systems engineering principle from the stake holder requirements to the system validation and verification phase results in a system that is consistent from the beginning to the end and that serves the intended purpose effectively when designed. Initial stake holder requirements are used to develop the use cases and scenarios which reflect the actual purpose of the system. From the use cases, system level requirements are generated which are then synthesized and broke down into low level requirements. These requirements are traced onto the use cases to ensure that all the scenarios and use cases are reflected in the requirements. System modelling and analysis results in the system structure and system behaviour. Requirements are allocated onto objects and attributes of system structure and methods of system behaviour. Qualitative and quantitative requirements are then assigned possible values and are known as specifications. The system trade-off analysis is carried out after identifying the measures of effectiveness and the parameters. This results in the best possible system design alternative among group of alternatives depending on the specified criteria.

The two important parts of this systems approach are System validation and verification, as they check the correctness of the system designed so far. Validation - "Are we building the right product? " and Verification - " Are we building the product, right?" Satisfactory answers to both the questions are a prerequisite to customer acceptance. A verification plan is proposed to test all the leaf level requirements. The tests, demonstrations, simulations and examinations defined in the verification plan are traced on to the requirements. Some of the tests, demonstrations, simulations and examinations are joined together and are assigned a verification string matrix which saves the cost and time involved in testing. Coverage and completeness criteria explains how the test plan covers all the requirements. If the system passes all the tests, it can be assured that the final system after actual implementation will meet the customer requirements. Thus, the application of systems engineering principles and methods result in effective system design with reduced costs and time.

REFERENCES

- Mark Austin, ENSE 623 Systems Validation and Verification Lecture Notes, Fall 2003.
- Mark Austin, A comprehensive Approach to Requirements Engineering, Short Course Notes, September, 2003.
- Mark Austin, Systems Engineering Validation and Verification, Reading for ENSE 623/ENPM 643, Vol 1, 2 and 3.
- Design Structure Matrices Tutorial, Mark Austin, http://www.isr.umd.edu/~austin/nsf-crcd/ds-matrices.html
- O'Grady J.O., System Validation and Verification, CRC press, 1988.
- CPLEX Online Manual http://www.dmi.usherb.ca/laboratoires/documentations-logiciels/cplex75/cplex75/doc/homepage/manuals.html
- Multi Objective Optimization http://www.isr.umd.edu/Courses/BARAS-ENSE623/secured/Class%20Handouts/Trade-Off-1.pdf
- Design of High-Volume Web Servers, http://www.webguild.org/presentations/HV-Web-Sites.pdf
- Cost Analysis on Data Encryption Keys, http://digitalenterprise.org/security/key_length.html
- RSA Encryption – Cost vs. Strength, http://www.rsasecurity.com/
- Oracle and SQL Server Price Comparison, http://www.microsoft.com/sql/evaluation/compare/pricecomparison.asp